Calculation of quantum discord and entanglement measures using the random mutations optimization algorithm

A.Yu. Chernyavskiy*

Institute of Physics and Technology, Russian Academy of Sciences, 117218, Moscow, Russia and

Faculty of Computational Mathematics and Cybernetics,
Lomonosov Moscow State University, 119991, Moscow, Russia

The simple and universal global optimization method based on simplified multipopulation genetic algorithm is presented. The method is applied to quantum information problems. It is compared to the genetic algorithm on standard test functions, and also tested on the calculation of quantum discord and minimal entanglement entropy, which is an entanglement measure for pure multipartite states.

1. INTRODUCTION

Research in quantum information science often leads to global optimization problems, and one of the main examples of this is the calculation of numerical measures of quantum correlations. One class of such measures is entanglement measures of pure multipartite quantum states. For instance, the calculation of the geometric entanglement measures for pure states [1] and the calculation of the minimal measurement entropy [2], which is considered in this paper, represents the global optimization problems. The second one is measures of mixed states. Most of such measures are based on minimization. Sometimes it's possible to use special optimization methods like convex optimization [3], but in most cases we need to use general algorithms. For example, global minimization is needed to calculate the relative entropy of entanglement [4], geometric measures of entanglement [1], any convex roof of measure for pure states (for example, the entanglement of formation [5]).

The third type of quantum correlation measures is quantum discord, which was proposed independently by [6] and [7]. Quantum discord is based not on entanglement, but on the existence of non-commutative quantum measurements, so separable (not entangled) states can have non-zero discord. There is a supposition that not only entanglement, but also quantum discord can be the resource for computations that outperform classical ones [8]. The definition of quantum discord is based on the minimization, and the analytical calculations was made only for a few classes of quantum states [9–11], that's why it's perspective to use modern global optimization algorithms.

It's well known that there is no universal method for finding global extremum of an arbitrary function. Till last years gradient methods (Newton's method, gradient descent, conjugate gradient method, etc.) were the most popular for optimization problems [12]. However, these methods are adapted for finding a local extremum or a global extremum of very specific functions.

Nevertheless, recently the increasing computational power allows to use more resource-intensive, but more effective algorithms. Generally, such algorithms use biological or physical heuristics. The most popular algorithms are ant colony optimization [13], genetic algorithm [14], particle swarm optimization [15], quantum annealing [16] and simulated annealing [17].

We presents random mutations algorithm, which is the simplified genetic algorithm. The algorithm is being tested on standard global optimization problems and on the computation of the minimal measurement entropy, which is the entanglement monotone for pure multipartite states [2, 18]. After that the algorithm is used to compute quantum discord and tested on two-qubit states with maximally mixed subsystems.

---

*Electronic address: andrey.chernyavskiy@gmail.com

2. RANDOM MUTATIONS OPTIMIZATION

2.1. General algorithm

Let's describe the random mutations minimization algorithm.

Consider a general real global optimization problem. For a given n-parameter function $f(\mathbf{x})$ (fitness function) we need to find a vector $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ such that

$$f(\mathbf{x}) = \min_y f(\mathbf{y}). \tag{2.1}$$

The general random mutations algorithm consists of the following steps:

1. **Initialization.** Generate n_{pop} random real vectors x_i of dimension n, $i \in \{\overline{1, n_{pop}}\}$.

2. **Mutation**. From each x_i generate a set D_i of n_{des} vectors of dimension n (descendants of x_i). Each vector of D_i is independently generated from x_i by a probabilistic mutation process. We can also include x_i into D_i.

3. **Selection.** From each set D_i we choose one "winner" vector and take it as x_i for the next generation. Also we store the best one of new x_i vectors (the vector with minimal f) as x_{best}.

4. **Termination.** If the termination conditions are satisfied, we take x_{best} as the result, else go to the step 2.

As it can be noticed, this algorithm is very similar to multipopulation genetic algorithm without migration and crossover.

The most important and specific steps are mutation (2) and termination conditions (4). Let's describe all steps.

2.2. Algorithm steps

2.2.1. Initialization

The process of the generation of initial vectors is probabilistic. We can choose different distributions according to a minimization problem. But almost always we can take uniform distribution on $[v_{min}, v_{max}]$ for each component of x_i.

2.2.2. Mutation

We use some special adaptive type of mutation. First of all, we must choose the distribution NM for the number of components being mutated. For example we can choose uniform distribution on $[1, n_{maxmut}]$. This is very important parameter of algorithm and, of course, must be chosen according to the length of the vector.

Every new vector $x_{ij} \in G_i$ is generated from x_i by the following procedure: we randomly choose $n_{mut} \sim NM$ components of the vector x_i and then independently change each of them by the rule

$$v_{new} = v_{old} + m \cdot b^p, \tag{2.2}$$

where m (magnitude) and p (power) are uniformly distributed on $[-1, 1]$ and $[p_{min}, p_{max}]$ respectively. The p_{max} parameter always depends on the minimization problem and is important. We usually take $p_{min} = -9$ and the base $b = 10$.

Such type of mutation is very similar to decimal or binary coding in genetic algorithms. It helps to adapt mutation amplitude during the algorithm: in the beginning even big mutations can win, but when the algorithm is near minimum almost only small steps can decrease the target function. But it's important that even during the final steps algorithm can find an appropriate large mutation (this is good for challenging local minima problem).

2.2.3. Selection

We use the simplest selection process: the vector with minimal fitness is taken as a winner. Another selection can be used, for example, winner can be chosen probabilistically according to its fitness.

2.2.4. Termination conditions

Every standard termination condition can be used, but the best among experiments was the next one: the algorithm stops if for every of n_{stall} last generations the x_{best} was changed on the value lesser than *eps*.

This termination condition is very effective against local minimum problem.

2.3. Features and parameters

The main advantages of random mutations algorithm are simplicity and easy parallelization. As a consequence the algorithm can be effectively realized and used on high-performance distributed computing systems. As it will be shown further, in spite of its simplicity, random mutations can outperform more complex algorithms.

The main parameter influencing the work of the algorithm is n_{pop}. Obviously, increase of n_{pop} leads to higher convergence speed, but also increases the probability to converge to local extrema. The second important parameter is the number of descendants n_{des}. Like population size parameter of genetic algorithms it strongly depends on the optimization problem and in most cases it must be tuned manually to obtain better results. The parameters n_{maxmut} and p_{max} can be chosen accordingly to the number of arguments and the scale of a fitness function.

Notice that n_{pop} subpopulations of random mutations algorithm are not totaly independent, because they are linked by termination conditions. That's why the algorithm is not similar to the simple repeating of the single population optimization procedure many times. Some subpopulations can converge to "bad" local minimums, near which the convergence is slow, and termination conditions stops the algorithm in such cases.

3. TESTING ON STANDARD FUNCTIONS

Random mutations algorithm was tested on different standard test functions [19, 20]:

$$F_{Rosenbrock}(x) = \sum_{i=1}^{n-1} \left(100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2\right),$$

$$\min_{-2.048 \leq x_i < 2.048} F_{Rosenbrock}(x) = F(1, 1, \ldots, 1) = 0;$$

(3.1)

$$F_{Rastrigin}(x) = 10n + \sum_{i=1}^{n} \left(x_i^2 - 10\cos(2\pi x_i)\right),$$

$$\min_{-5.12 \leq x_i < 5.12} F_{Rastrigin}(x) = F(0, 0, \ldots, 0) = 0;$$

(3.2)

$$F_{Griewank}(x) = 1 + \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \left(\cos(\frac{x_i}{\sqrt{i}})\right),$$

$$\min_{-512 \leq x_i < 512} F_{Griewank}(x) = F(0, 0, \ldots, 0) = 0;$$

(3.3)

$$F_{Schwefel}(x) = 418.98288727 \cdot n - \sum_{i=1}^{n} x_i \sin\left(\sqrt{|x_i|}\right),$$

$$\min_{-512 \leq x_i < 512} F_{Schwefel}(x) = F(420.968750, \ldots, 420.968750) = 0.$$

(3.4)

Of course, the assessment of the dependence of the accuracy from the number of algorithm iterations is not good, because the single iteration can be very difficult. So, we need to research the dependency of the accuracy from the number of fitness function evaluations. The main task of standard tests was only the verification of the correctness and competitiveness of the random mutations algorithm. The test functions was taken with the number of parameters $n = 50$. We used the standard MATLAB genetic algorithm. The size of the population was 5000, other parameters was standard. The parameters for random mutations were as follows: $n_{pop} = 40$, $n_{maxmut} = 5$, for the Rosenbrock
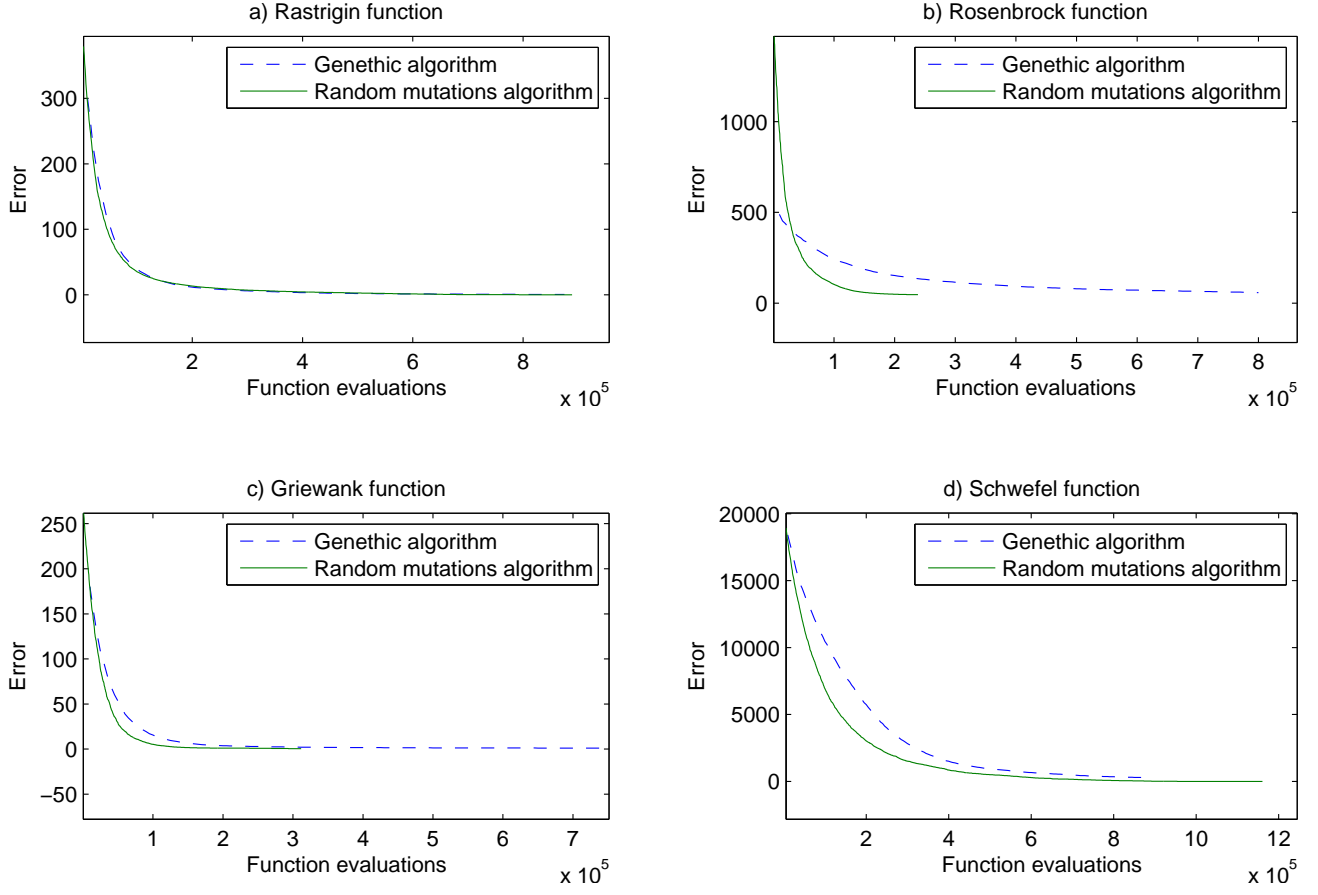
Figure 1: The convergence of the random mutations and genetic algorithm on standard test problems.

and Schwefel functions $n_{des} = 20$, for the Rosenbrock and Griewank functions $n_{des} = 10$. For both algorithms $n_{stall} = 50$, $\varepsilon = 10^{-6}$, but algorithms have different termination conditions. The MATLAB genetic algorithm stops if the averaged change of the best fitness over last n_{stall} generations is lesser than eps. The test results averaged over 20 experiments are plotted on Fig. 1. The graphs for each algorithm are presented for the minimal number of steps over experiments. We take $F_{Griewank}(400 \cdot x)$ to preserve scale The constraints of Schwefel function (as opposed to other test functions) is significant. Moreover the global minimum is closer to the bound than to the center of the search space, so the scaling begin to influence results. We take $F_{Schwefel}(350 \cdot x)$ for genetic algorithm and $F_{Schwefel}(100 \cdot x)$ for random mutations. The best and average results over experiments are presented on Tab. I.

	Rastrigin	Rosenbrok	Griewank	Schwefel
Genetic algorithm	$f_{best} = 0.0781$ $f_{avg} = 0.1983$	$f_{best} = 37.7922$ $f_{avg} = 49.8762$	$f_{best} = 0.3377$ $f_{avg} = 1.0207$	$f_{best} = 18.4127$ $f_{avg} = 53.9818$
Random mutations	$f_{best} = 0.00002$ $f_{avg} = 0.00006$	$f_{best} = 32.7622$ $f_{avg} = 42.4590$	$f_{best} = 0.0194$ $f_{avg} = 0.1171$	$f_{best} = 0.0006$ $f_{avg} = 0.0007$

Table I: The best and average results of genetic and random mutations algorithm on test functions over 20 experiments.

This method of assessment is good for the test functions, which have regular structure. However, it is not so good for some real tasks. Consider a function with a local minima "plateau" and a global minima with a close value. The sample of such function is illustrated on Fig. 2.

Global optimization algorithms are probabilistic and with large probability converge to local minimums of such functions. But the probability of the correct answer is nonzero. So we need to assess the dependence of this probability from the number of fitness function evaluations. The exact method of such assessment will be described in Sec. 4.3.
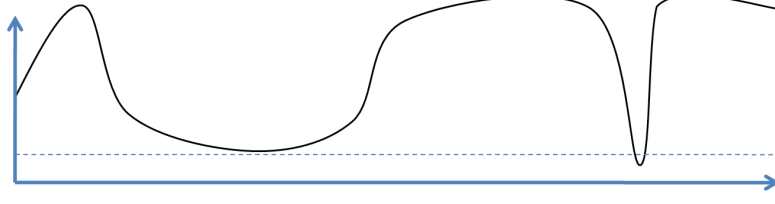
Figure 2: Function which is difficult for finding the global minimum.

4. CALCULATION OF MINIMAL MEASUREMENT ENTROPY FOR PURE STATES

4.1. Definition

Let

$$|\psi\rangle = \sum_{i_1,i_2,\ldots,i_n=0}^{1} \lambda_{i_1 i_2 \ldots i_n} |i_1 i_2 \ldots i_n\rangle \tag{4.1}$$

be a pure quantum n-qubit state (the definition for qudits is the same).

The real positive number

$$H_{meas}(|\psi\rangle) = - \sum_{i_1,i_2,\ldots,i_n=0}^{1} |i_1 i_2 \ldots i_n|^2 log|i_1 i_2 \ldots i_n|^2 \tag{4.2}$$

is called a *measurement entropy.*

Then the minimal measurement entropy over all orthogonal local measurement basis sets

$$E_{Hmin}(|\psi\rangle) = \min_{U_1,U_2,\ldots,U_n} H_{meas}(U_1 \otimes U_2 \otimes \ldots \otimes U_n|\psi\rangle) \tag{4.3}$$

is an entanglement measure (monotone) [2]. Here U_i are unitary matrices.

4.2. Calculation

For an arbitrary state 4.1 we need to find the global minima of $H_{meas}(U_1 \otimes U_2 \otimes \ldots \otimes U_n|\psi\rangle)$ over all possible sets of the 2×2 unitary matrices $U_1, U_2, \ldots, U_n$.

The parametrization of 2×2 unitary matrices is well known:

$$U(\alpha,\beta,\delta,\gamma) = e^{i\alpha} \begin{pmatrix} e^{i(-\beta-\delta)} \cos\gamma & -e^{i(-\beta+\delta)} \sin\gamma \\ e^{i(\beta-\delta)} \sin\gamma & e^{i(\beta+\delta)} \cos\gamma \end{pmatrix}, \tag{4.4}$$

where β,δ,γ are real numbers. We can easily see that α and β doesn't affect H_{meas}, so the final parametrization for our task is

$$U(\delta,\gamma) = \begin{pmatrix} e^{-i\delta} \cos\gamma & -e^{i\delta} \sin\gamma \\ e^{-i\delta} \sin\gamma & e^{i\delta} \cos\gamma \end{pmatrix}, \tag{4.5}$$

and we need to solve the global minima problem for $2n$ real parameters.

If we want to calculate E_{Hmin} for qudits ($d > 2$) we can use the parametrization $U = e^{iH}$, where H is Hermitian and can be obviously parameterized by d^2 real numbers.

4.3. Testing and results

The main task of this work is testing of the random mutations algorithm, so we will present only test results, more calculations of E_{Hmin} can be found in [2, 18].

The validity of calculations was tested on four cases:

1. **Unentengled states.** Of course, for unentangled states minimal measurement entropy is equal to zero:

$$E_{Hmin}(U_1 \otimes U_2 \otimes \ldots \otimes U_n |00 \ldots 0\rangle) = 0. \tag{4.6}$$

2. **GHZ states.** For the generalized GHZ-states $|GHZ\rangle = \lambda_0 |00 \ldots 0\rangle + \lambda_1 |00 \ldots 0\rangle$:

$$E_{Hmin}(|GHZ\rangle) = H_{sh}(\lambda_0, \lambda_1) = -|\lambda_0|^2 \log |\lambda_0|^2 - |\lambda_1|^2 \log |\lambda_1|^2. \tag{4.7}$$

3. **Bipartite states.** For bipartite states E_{Hmin} equals reduced von Neumann entropy.

4. **Additivity.** For arbitrary n-qubits state $|\varphi\rangle$ and k-qubits state $|\psi\rangle$ for the $(n+k)$-qubits state $|\varphi\rangle \otimes |\psi\rangle$:

$$E_{Hmin}(|\varphi\rangle \otimes |\psi\rangle) = E_{Hmin}(|\varphi\rangle) + E_{Hmin}(|\psi\rangle). \tag{4.8}$$

For all tests the random mutations algorithm gives correct results (tests were made for up to 17 qubits, the accuracy was 10^{-5}). For these test states the algorithm shows a very good convergence. But some other states were very difficult, because the corresponding fitness function is of the type described in 3 (with local minima "plateau"). For example, for one state of 7-qubit Grover's algorithm the most possible result of the algorithm is 4.0220, but there is another minimum 3.968. We suppose that the second one is global, because of large amount of tests. Let's describe the method of quality assessment of global optimization algorithms on such problems, which was claimed in Sec. 3.

Consider n_{exp} numerical experiments. It's obvious that the probability of the error decreases exponentially with repetition of optimization and we can take up the mean value of fitness calculations for achieving the success probability rate 0.5.

$$e_{0.5} = -n_{evaluations}/n_{exp} \cdot log_{0.5}(n_{err}/n_{exp}), \tag{4.9}$$

where $n_{evaluations}$ is the summarized number of fitness function evaluations during all experiments, n_{err} is the number of incorrect results. I.e. $n_{evaluations}/n_{exp}$ is the average number of fitness function evaluations during one experiment, n_{err}/n_{exp} is the probability of error.

As there is no analytical values of E_{Hmin} for computationally difficult states, we take the best result after 10000 repeats of genetic, swarm particle, and random mutations algorithm for the "correct" global minimum.

Here is the example results for 17th step of 11-qubits Grover's algorithm.

- genetic algorithm: $e_{0.5} \approx 5500$

- swarm particle optimization: $e_{0.5} \approx 11000$

- random mutations: $e_{0.5} \approx 3500$.

Other hard states give similar results.

Random mutations algorithm is very simple, but it demonstrates significant improvement for such problems. We must note that only a very simple type of swarm optimization was used, so different types and topologies probably may show better results.

Described tests show us that we can use random mutations algorithm to solve some difficult optimization problems of quantum information theory.

5. CALCULATION OF QUANTUM DISCORD

5.1. Definition

Discord is a measure of non-classical correlations in quantum systems. Consider the bipartite mixed quantum state ρ. Discord represents the difference between quantum mutual information and classical correlations:

$$D(\rho) = I(\rho) - C(\rho). \tag{5.1}$$

The $I(\rho) = S(\rho_A) + S(\rho_B) - S(\rho)$ is the quantum mutual information. Here $S(\rho) = Tr\rho \log \rho$ is von Neumann entropy, ρ_A and ρ_b are reduced density matrices of subsystems.

The equation for classical correlations $C(\rho)$ is not so simple. First of all it's not symmetric. Let's choose subsystem B and make some projective measurement with the observable M, this measurement will destruct quantum correlations. Classical correlations after the measurement will be defined by $C_M(\rho) = S(\rho_A) - S(\rho|M)$, where $S(\rho|M) = \sum_i p_i S(\rho_i)$, is a quantum conditional entropy, p_i and ρ_i are the probabilities and results of the measurement M. To calculate $C(\rho)$ and discord itself we need to get the maximal $C_M(\rho)$, so $C(\rho) = S(\rho_A) - \min_M S(\rho|M)$.

5.2. Testing

We use random mutations optimization to calculate quantum discord. To do it we need to parameterize an observable M. The simplest way is to parameterize $n \times n$ hermitian matrix by n^2 real numbers: n on the diagonal and $n(n-1)$ above the diagonal. This method is redundant, because we only need the projectors of M, but not its eigenvalues, but this redundancy only adds some additional work for the optimization algorithm, but it obviously doesn't affect the result.

The analytical results of Luo [9] were used to test the possibility of the random mutations algorithm to compute quantum discord of two-qubit systems . Consider a state

$$\rho = \frac{1}{4}(I + \sum_{j=1}^{3} c_j \sigma_j \otimes \sigma_j). \tag{5.2}$$

This is a state with maximally mixed subsystems. The quantum mutual information of this state is

$$I(\rho) = 2 + \sum_{i=0}^{3} \lambda_i \log_2 \lambda_i, \tag{5.3}$$

where

$$\lambda_0 = \tfrac{1}{4}(1 - c_1 - c_2 - c_3), \quad \lambda_1 = \tfrac{1}{4}(1 - c_1 + c_2 + c_3),$$

$$\lambda_2 = \tfrac{1}{4}(1 + c_1 - c_2 + c_3), \quad \lambda_3 = \tfrac{1}{4}(1 + c_1 + c_2 - c_3) \tag{5.4}$$

are the eigenvalues of ρ.

The constraints of the coefficients c_j are such that $0 \le \lambda_j \le 1$.

For the classical correlations:

$$C(\rho) = \frac{1}{2}[(1 - c) \log_2(1 - c) + (1 + c) \log_2(1 + c)], \tag{5.5}$$

where $c = max\{c_1, c_2, c_3\}$.

By definition, $D(\rho) = I(\rho) - C(\rho)$. It's important to notice that some important quantum spin systems can be described by 5.2, so the value of discord for these systems can be calculated analytically [21].

Let's describe tests carried out on such states. We simply generate random sets $\{c_1, c_2, c_3\}$, where each c_j is uniformly distributed on $[-1, 1]$. If the corresponding density matrix is valid we calculate discord for it and compare its value with the analytical result. More than 5000 states was checked. The parameters of algorithm was as follows: $n\_mut = 5$, $n\_pop = 10$, $n\_des = 10$. In all experiments the desired accuracy was achieved: difference between analytical and numerical results was lesser then 10^{-6}.

6. CONCLUSIONS

The random mutations global optimization algorithm was presented, which is a kind of simplified multipopulation genethic algorithm. It gives the correct results and outperforms simple genetic optimization on standard tests and on the computation of minimal measurement entropy, which is the entanglement monotone for multipartite pure states. The algorithm was also used to compute quantum discord and tested on analytically known discord values of two-qubit states with maximally mixed subsystems. Thus, we can conclude that random mutations algorithm is perspective to solve some difficult problems of quantum information theory like calculation of discord and different entanglement measures for multipartite states.

Acknowledgments

[1] T.C. Wei and P.M. Goldbart. Geometric measure of entanglement and applications to bipartite and multipartite quantum states. *Physical Review A*, 68(4):42307, 2003.

[2] A Yu Chernyavskiy. Entanglement measure for multipartite pure states and its numerical calculation. *arXiv preprint arXiv:0905.0201*, 2009.

[3] Koenraad Audenaert, J Eisert, E Jané, MB Plenio, S Virmani, and Bart De Moor. Asymptotic relative entropy of entanglement. *Physical review letters*, 87(21):217902, 2001.

[4] L Henderson and V Vedral. Information, relative entropy of entanglement, and irreversibility. *Physical review letters*, 84(10):2263–2266, 2000.

[5] W.K. Wootters. Entanglement of formation of an arbitrary state of two qubits. *Physical Review Letters*, 80(10):2245–2248, 1998.

[6] Harold Ollivier and Wojciech H Zurek. Quantum discord: a measure of the quantumness of correlations. *Physical review letters*, 88(1):017901, 2001.

[7] Leah Henderson and Vlatko Vedral. Classical, quantum and total correlations. *Journal of Physics A: Mathematical and General*, 34(35):6899, 2001.

[8] Animesh Datta, Anil Shaji, and Carlton M Caves. Quantum discord and the power of one qubit. *Physical review letters*, 100(5):050502, 2008.

[9] Shunlong Luo. Quantum discord for two-qubit systems. *Physical Review A*, 77(4):042303, 2008.

[10] Mazhar Ali, ARP Rau, and Gernot Alber. Quantum discord for two-qubit x states. *Physical Review A*, 81(4):042105, 2010.

[11] CC Rulli and MS Sarandy. Global quantum discord in multipartite systems. *Physical Review A*, 84(4):042109, 2011.

[12] J. Nocedal and S.J. Wright. *Numerical optimization*. Springer, 1999.

[13] M. Dorigo. Optimization, learning and natural algorithms. *Milano: Politecnico di Italy, doktorska disertacija*, 1992.

[14] JH Holland. Adaptation in natural and artificial system: an introduction with application to biology, control and artificial intelligence. *Ann Arbor, University of Michigan Press*, 1975.

[15] J.F. Kennedy. *Swarm intelligence*. Springer, 2001.

[16] B. Apolloni, C. Carvalho, and D. De Falco. Quantum stochastic optimization. *STOCHASTIC PROCESS. APPLIC.*, 33(2):233–244, 1989.

[17] S. Kirkpatrick. Optimization by simulated annealing: Quantitative studies. *Journal of Statistical Physics*, 34(5):975–986, 1984.

[18] A Yu Chernyavskiy. *Minimal measurement entropy as a computable entanglement measure of multipartite quantum states (in Russian)*. PhD thesis, Institute of Physics and Technilogy of RAS, 2010.

[19] L. Panait and S. Luke. A comparative study of two competitive fitness functions. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002)*. Citeseer, 2002.

[20] S. Janson and M. Middendorf. A hierarchical particle swarm optimizer and its adaptive variant. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 35(6):1272–1282, 2005.

[21] EB Feldman, EI Kuznetsova, and MA Yurishchev. Quantum correlations in a system of nuclear s= 1/2 spins in a strong magnetic field. *Journal of Physics A: Mathematical and Theoretical*, 45(47):475304, 2012.